MDPI

Article

Benchmarking Summarization Methods for Scientific Abstracts: From Classical Models to LLMs

Fabio Baumgärtel [1], Enrico Bono [1,2], Paul Perco [1,3] and Matthias Ley [1,2]*

[1] Delta4 GmbH, Vienna, Austria
[2] Division of Pediatric Nephrology and Gastroenterology, Department of Pediatrics and Adolescent Medicine, Comprehensive Center for Pediatrics, Medical University Vienna, Vienna, Austria
[3] Department of Internal Medicine IV, Medical University Innsbruck, Innsbruck, Austria
* Correspondence: matthias.ley@delta4.ai

Abstract

A single paragraph of about 200 words maximum. For research articles, abstracts should give a pertinent overview of the work. We strongly encourage authors to use the following style of structured abstracts, but without headings: (1) Background: place the question addressed in a broad context and highlight the purpose of the study; (2) Methods: describe briefly the main methods or treatments applied; (3) Results: summarize the article's main findings; (4) Conclusions: indicate the main conclusions or interpretations. The abstract should be an objective representation of the article, it must not contain results which are not presented and substantiated in the main text and should not exaggerate the main conclusions.

Keywords: benchmarking; natural language processing; text summarization; largue language models

1. Introduction

With the exponential growth of publicly available data, the effort to properly select relevant information has increased dramatically, leading to a scenario of information overloading in which important data may remain hidden. This has created a need for the development of reliable tools that can efficiently generate high-level summaries to highlight only the essential parts. Particularly, in the scientific field, where finding the right content is crucial for generation of novel hypothesis. To address this necessity, automatic text summarization (ATS) methods have undergone significant advances over time, enhancing their reliability in accurately summarizing relevant parts of complex research articles. While the history of ATS has been extensively evaluated and described by several articles [1,2], only few of them are tailored on scientific literature summarization [3,4]. This introduction, therefore, aims to cover all the methodological improvements in ATS, ranging from early statistical approaches to modern large language models (LLM), by linking each method to its scientific application.

1.1. Pre-Neural era: from word-frequency and early statistical methods to graph based approaches

Pre-neural era of summarization of scientific literature was characterized mainly by extractive approaches, where in an unsupervised way, summaries were generated by using word or concept frequency to identify relevant sentences. The first word-frequency based approaches were discussed in the Luhn's paper [5],which presented a method based on the assumption that recurrent words in a text are likely more important and Edmunson

Received:
Revised:
Accepted:
Published:

Citation: . Title. *Int. J. Mol. Sci.* **2025**, *1*, 0. https://doi.org/

[6], who introduce concepts as cue words, title words, and sentence position to enhance the automatic summarization process. Later, the TD-IDF method, Term Frequency–Inverse Document Frequency, was developed [7]. This approach was applied to text summarization, where sentences were represented by term-weight vectors, with weights assigned to words to down-weight common biomedical terms that are frequent but less relevant, and up-weighting rare terms that might be more relevant. Thus, word-frequency based approaches have been extensively adopted in scientific text summarization, being at the basis of more sophisticated strategies [8]. Afterwards, graph-based methods were adopted, where sentences were represented as nodes and relations between sentences, calculated by using similarity measures (i.e cosine similarity of TF-IDF vectors), as edges. Two Graph-based methods gained popularity in the biomedicals domain: i) TextRank, that build a graph by breaking down the documents into single sentences and then exploit the PageRank algorithms to assign importance to each sentence, ultimately building the summary by using the top ranked ones [9,10] and ii) LexRank that instead, use eigenvector centrality to find the ones that are most influential in the graph. [11]

1.2. Neural network era: from sequence-to-sequence frameworks to modern LLM

With the advent of Sequence-to-Sequence framework (Seq2seq), summaries were generated by paraphrasing and condensing the text. This involves the use of an Encoder-Decoder architecture, originally implemented as Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs) and then adapted to the biomedical domain [12,13]. Later, the introduction of self-attention mechanisms led to the replacement of recurrent networks, able to process sequencies in a sequential way, in favours of self-attention mechanisms, that handle entire sequence in parallel, capturing more complex linguistics patterns and long context relationship [14]. This approach is at the basis of Transformer that gained popularity for performing different biomedical Natural Language Processing (NLP). One of the earliest language representation models, was the BERT model, a Bidirectional Encoder Representations from Transformers [15]. This type of architecture has been widely adopted in the biomedical field thanks to the possibility of being fine-tuned by adding a task-specific output layer. Of note, for scientific text summarization BERT was coupled with unsupervised clustering methods to cluster the produced embeddings based on similarity, ultimately ranking the most relevant sentences to generate the summary [16–18]. Building on this, models for abstractive summarization inspired by the BERT architecture appeared: i) BART, Bidirectional and Auto-Regressive Transformer, a denoising autoencoder for pretraining sequence-to-sequence models [19], which can be adapted to the scientific literature by pre-training or fine-tuning the model on biomedical text [20,21]. ii) T5, Text-to-Text Transfer Transformer, was introduced as a unified text-to-text framework for different NLP task. The advantage of using this model relies on its flexibility, as there is no need to architectural changes to be used for specific tasks [22]. iii) PEGASUS, Pre-training with Extracted Gap-sentences for Abstractive Summarization Sequence-to-sequence models, was proposed as a specific model designed for abstractive summarization [23]. Of note, some fine-tuned versions are “google/pegasus-pubmed” and “google/bigbird-pegasus-large-pubmed” that combine BigBird [24] and PEGASUS. iiii) Longformer [25],built on RoBERTa, along its Encoder-Decoder variant for text-to-text generation “allenai/led-base-16384” that was built to handle longer text. Particularly, “led-large-16384-arxiv”, a model fine tuned on arXiv data was developed [26]. However, despite these advances, the field of ATS quickly moved towards the use of LLMs, decoder-only transformer architecture, which represent more flexible and performant models in capturing semantic relations. LLMs can be generally classified in i) general-purpose LLMs, which leverage their broad training on different domains dataset

to generate abstractive summaries, reasoning- oriented LLMs, which are fine-tuned for multi-step reasoning through advanced chain-of-thought and instruction tuning to increase general understanding [27] and domain-specific LLMs, designed to specifically address the challenges of complex biomedical corpus. Several families of LLM have been developed over the years. The OpenAI GPT series models, from the first GPT-1 [28] to the recently released reasoning- oriented GPT 5 series (Nano, Mini, Full) and GPT:OSS, are all pre-trained using a huge amount of data in a self-supervised way. This family includes some domain-specific variant as BioGPT [29], specifically fine-tuned on the biomedical domain. Similarly, Anthropic Claude Models are built on transformer architecture trained by using an approach called Constitutional AI [30] including also some reasoning models: Claude Sonnet-4, Opus-4 and Opus-4-1. The Meta Llama family, with Llama 3 being the most capable model of the family, has also developed some domain-specific adaptations such as OpenBioLLM-Llama-3, a variant of Meta's Llama-3 trained on a large corpus of high-quality biomedical data and medllama2, a medical language model built on Meta's LLaMA 2 architecture. Google and Microsoft developed a series of lightweight models: the Google Gemma series [31], which includes Gemma3 as its latest and most powerful reasoning model and the Microsoft Phi series, which comprises Phi-4-reasoning and Phi-4-mini-reasoning. Other models such as Granite 4.0, a reasoning model of the IBM's Granite series and Magistral, the first reasoning model of Mistral have also been released. Additionally, Mistral developed Biomistral, an open-source model pretrained on PubMed Central data. Moreover, Alibaba Cloud's introduced the Qwen 3 series as an open-source LLM family, where recently, SciLitLLM has been further developed as a specialized model for scientific literature understanding based on Qwen2.5 and trained through continual pre-training (CPT) and supervised fine-tuning (SFT) on scientific literature [32]. Lastly, DeepSeek has also developed some reinforcement learning (RL)-driven reasoning models, which are cost-effective and efficient [33].

At the best of our knowledge, no comprehensive peer-reviewed publications that asses the performance of all the discussed methods are available. Therefore, our work, aim at evaluating the efficiency of most of the discussed approaches, highlighting both strengths and limitations of different methods and providing useful insights in using these tools for boosting knowledge discovery in molecular sciences.

2. Materials and Methods

2.1. Gold-Standard Dataset

To establish a reliable benchmark for automatic summarization, we assembled a gold-standard dataset of 1,000 biomedical articles drawn from a diverse set of peer-reviewed journals hosted on *ScienceDirect* and *Cell Press*. These journals were selected because, in addition to their focus on molecular and biomedical sciences, they provide a standardized *Highlights* section [34,35]. This section provides concise bullet points that capture the main findings of each article. These served as the reference summaries in our evaluation, while the corresponding abstracts were used as input texts for the summarization.

Articles were collected systematically across a variety of journals to ensure coverage of different fields within molecular sciences such as drug discovery, genomics, proteomics, biotechnology, and biochemistry. We selected 50 articles from each of the 20 journals, bringing the dataset to 1,000 in total. The distribution of articles across journals is summarized in Table 1.

Table 1. Overview of journals and number of articles included in the gold-standard dataset.

Publisher	Journal
ScienceDirect	Drug Discovery Today
ScienceDirect	Journal of Molecular Biology
ScienceDirect	FEBS Letters
ScienceDirect	Journal of Biotechnology
ScienceDirect	Gene
ScienceDirect	Genomics
ScienceDirect	Journal of Proteomics
ScienceDirect	The International Journal of Biochemistry & Cell Biology
ScienceDirect	Cytokine
ScienceDirect	Developmental Cell
Cell	Cell
Cell	Cancer Cell
Cell	Cell Chemical Biology
Cell	Cell Genomics
Cell	Cell Host & Microbe
Cell	Cell Metabolism
Cell	Cell Reports
Cell	Cell Reports Medicine
Cell	Cell Stem Cell
Cell	Cell Systems

This setup provides standardized pairs of abstracts and reference summaries that can be directly used for evaluating automatic summarization methods.

2.2. Summarization Methods

We evaluated 63 summarization models, ranging from simple frequency-based algorithms to state-of-the-art large language models (LLMs). By having this extensive coverage of models, we were able to compare established techniques with the latest transformer-based models under identical conditions.

The models were grouped into five categories:

1. Traditional models: As a foundation for comparison, we included two traditional extractive models: a simple frequency-based approach and TextRank [9]. These models provide a simple baseline to compare the more complex approaches with.
2. Encoder-Decoder models: We included a set of pre-trained encoder-decoder models, which are available through the HuggingFace library: BART (base and large) [19], T5 (base and large) [36], mT5 [37], and a variety of PEGASUS models [23]. These models are often applied for abstractive summarization and represent well-established neural systems within our benchmark.
3. General-purpose LLMs: We also evaluated a range of widely used large language models designed for broad application. This group includes models such as Gemma [31], Granite [38], LLaMA [39], Mistral [40], Phi [41,42], GPT [43,44], Claude [45], and Apertus [46], which represent the current landscape of general-purpose systems.
4. Reasoning-oriented LLMs: We further included several models developed with a focus on advanced reasoning capabilities. This group includes models from the DeepSeek-R1 family [47], Qwen [48], more GPT models such as GPT-oss [49] and GPT-5 [50], Magistral [51], and some additional Claude models. Their design emphasizes multi-step problem solving and allowed us to explore whether reasoning affects summarization performance.
5. Scientific/Biomedical models: To assess whether domain adaptation improves summarization quality, we included PEGASUS and BigBird models fine-tuned on PubMed

data (pegasus-pubmed & bigbird-pegasus-large-pubmed), LED [25] (arXiv-tuned), BioGPT [52], MedLLaMA2 [53], OpenBioLLM [54], BioMistral [55], and SciLitLLM1.5 models [56], which are trained on medical/biomedical data or on summarization tasks themselves.

The complete list of models included in each category is shown in Table 2.

Table 2. Overview of summarization methods/models evaluated in this study, organized by category.

Group	Methods/Models
Traditional models	textrank; frequency
Encoder-Decoder models	facebook/bart-base; facebook/bart-large-cnn; google-t5/t5-base; google-t5/t5-large; csebuetnlp/mT5_multilingual_XLSum; google/pegasus-xsum; google/pegasus-cnn_dailymail; google/pegasus-large
General-purpose LLMs	gemma3:270M; gemma3:1b; gemma3:4b; gemma3:12b; PetrosStav/gemma3-tools:4b; granite3.3:2b; granite3.3:8b; granite4:tiny-h; granite4:small-h; granite4:micro; granite4:micro-h; llama3.1:8b; llama3.2:1b; llama3.2:3b; mistral:7b; mistral-nemo:12b; mistral-small3.2:24b; mistral-small-2506; mistral-medium-2505; mistral-large-2411; mistral-medium-2508; phi3:3.8b; phi4:14b; gpt-3.5-turbo; gpt-4o; gpt-4o-mini; gpt-4.1; gpt-4.1-mini; claude-3-5-haiku-20241022; chat_swiss-ai/Apertus-8B-Instruct-2509
Reasoning-oriented LLMs	deepseek-r1:1.5b; deepseek-r1:7b; deepseek-r1:8b; deepseek-r1:14b; qwen3:4b; qwen3:8b; gpt-oss:20b; gpt-5-nano-2025-08-07; gpt-5-mini-2025-08-07; gpt-5-2025-08-07; claude-sonnet-4-20250514; claude-opus-4-20250514; claude-opus-4-1-20250805; magistral-medium-2509
Scientific/Biomedical models	google/pegasus-pubmed; google/bigbird-pegasus-large-pubmed; led_large_16384_arxiv_summarization; completion_microsoft/biogpt; medllama2:7b; chat_aaditya/OpenBioLLM-Llama3-8B; conversational_BioMistral/BioMistral-7B; chat_Uni-SMART/SciLitLLM1.5-7B; chat_Uni-SMART/SciLitLLM1.5-14B

With this selection, we covered models of different sizes and release periods, ensuring that both widely adopted systems and recent architectures were represented. Extraordinarily large models were not considered because their resource demands exceed what is practical for typical summarization pipelines and were beyond the resources available for this study.

These 63 diverse models were all tasked with generating summaries for each of the 1,000 abstracts in the dataset, resulting in 50,000 generated summaries available for evaluation.

2.3. Evaluation Metrics

As there is no single metric that can fully reflect summary quality, especially in the biomedical field where both coverage of key information and factual correctness are critical, we used a multitude of metrics grouped into three categories: traditional surface-level measures, embedding-based metrics, and performance-related measures that reflect the

feasibility of using the methods in real-world applications. By combining all these metrics into one final overall score, we end up with a balanced benchmark value that reflects both summary quality and practical usability.

2.3.1. Surface-level Metrics

This group consists of metrics that compare the generated summaries with the reference summaries mainly at the word or phrase level. While they do not capture meaning beyond surface overlap, they remain common metrics in summarization research and provide a simple foundation for evaluation. We used three ROUGE variants (ROUGE-1, ROUGE-2, ROUGE-L) [57], BLEU [58], and METEOR [59]. ROUGE-1 and ROUGE-2 measure how many unigrams (single words) or bigrams (word pairs) from the reference appear in the generated output, while ROUGE-L identifies the longest sequence of words shared between the two. BLEU calculates how many n-grams in the output also occur in the reference, but it emphasizes precision rather than recall and applies a brevity penalty to counteract the tendency toward overly short summaries. METEOR extends n-gram matching by also considering word stems and synonyms, which makes it more tolerant to variations in wording. Together, these metrics offer a simple but transparent point of reference.

2.3.2. Embedding-based Metrics

To capture similarity beyond surface-level word overlap, we included a set of embedding-based metrics built on pre-trained transformer models. These methods generate vector representations of text, which allows them to capture similarity in meaning rather than just word overlap. We employed RoBERTa [60] and DeBERTa [61], two transformer-based models with strong performance across natural language processing tasks. In the context of summarization evaluation, they can be used to judge whether two summaries capture the same content even if phrased differently.

We also included all-mpnet-base-v2 [62], a transformer model fine-tuned for sentence similarity. Unlike RoBERTa and DeBERTa, which are primarily general-purpose encoders, MPNet was trained with a focus on aligning at the sentence-level. This focus makes it a useful complement to the other metrics, as it is particularly sensitive to whether the overall sense of a reference summary is preserved in the system output.

Finally, to evaluate factual consistency, we applied AlignScore [63], a metric designed to test whether the statements in a generated summary are supported by the source text. In contrast to the other metrics, we used AlignScore in a way where it does not compare the output to the reference summary but instead aligns it directly with the abstract, as factual accuracy can only be judged relative to the original input. This addition ensures that our evaluation is sensitive to errors and hallucinations that might otherwise be overlooked.

2.3.3. Performance Metrics

In addition to summary quality, we also considered practical aspects of model performance. Four measures were included: output token cost reflects the average length of generated summaries in tokens, as excessively long outputs increase runtime and resource requirements. Insufficient findings describe how often a model returned the predefined token 'INSUFFICIENT_FINDINGS' instead of producing a summary, capturing cases where it concluded the input did not contain substantive findings. Acceptance is the proportion of prompts for which a model produced an output, since some models occasionally failed to return a response. Finally, speed records the average time required to generate summaries, which is critical when processing large datasets.

These measures complement the quality metrics by addressing whether a method is not only accurate but also feasible to use in practice.

2.4. Benchmarking Framework

The benchmark was conducted using Python 3.12. Gold standard data were retrieved from open-access publications published by ScienceDirect and Cell Press through manual extraction of titles, abstracts, and highlight sections, along with metadata including publication URLs, identifiers, section types, and article types where available. All data were stored in machine-readable JSON format.

The framework was implemented using the Python standard library supplemented by several specialized packages: pandas [64] for data import and export, scikit-learn [65] for computing cosine similarities of embeddings and TF-IDF vectors, networkx [66] for graph construction and PageRank algorithm [67]. Additional evaluation metrics were computed using NLTK [68] for METEOR and BLEU scores, ROUGE-score, BERT-score [69], AlignScore, and sentence-transformers [70] with the all-mpnet-base-v2 model.

Communication with proprietary closed-source LLMs was facilitated through the official Python APIs provided by Anthropic, Mistral AI, and OpenAI. Local LLM execution was performed on a workstation equipped with a NVIDIA RTX A4000 GPU (16GB VRAM) running Ollama as a backend service, accessed through its Python API along with the transformers library [71].

All LLMs were configured with a temperature parameter of 0.2 to optimize reproducibility while avoiding completely deterministic outputs. For the latest generation of OpenAI models featuring adaptive reasoning capabilities, the configuration was set to `text.verbosity = low` and `reasoning.effort = minimal`. The full set of parameters and prompts are documented in the `config.py` file in the repository.

2.5. Data Availability

The complete source code, documentation, gold standard dataset, and processed results are available at:
https://www.github.com/Delta4AI/LLMTextSummarizationBenchmark.

3. Results

Our benchmark results offer a comparative view of summarization performance across all evaluated models. We first present overall rankings, followed by comparisons between the main model groups. Additionally, we examine results on individual metrics, runtime performance, and correlations between the evaluation metrics used.

3.1. Overall Model Performance

Figure 1 provides an overview of the performance of all evaluated models across all surface-level and embedding-based metrics. Each row corresponds to one model, and each column to a specific metric, with lower ranks indicating better performance. Models are sorted by their average rank across metrics.

The best-performing models overall were from the Mistral family, with the top positions occupied by `ollama_mistral-small-3.2:24b`, `mistral_mistral-small-2506`, and `mistral_mistral_medium-2505`. Two OpenAI models (`gpt-5-nano-2025-08-07` and `gpt-5-mini-2025-08-07`) followed closely. These models achieved high ranks across nearly all surface-level metrics (ROUGE-1, ROUGE-2, ROUGE-L, METEOR, BLEU) and performed well on most embedding-based measures (RoBERTa, DeBERTa, all-mpnet-base-v2, AlignScore). Several other LLMs also achieved competitive scores and maintained stable rankings across metrics.

At the lower end of the ranking, encoder–decoder architectures such as T5 and PEGASUS, traditional extractive methods (TextRank and the frequency-based approach), and

scientific/biomedical models such as MedLLaMA2 and BioGPT, achieved lower scores on most metrics.

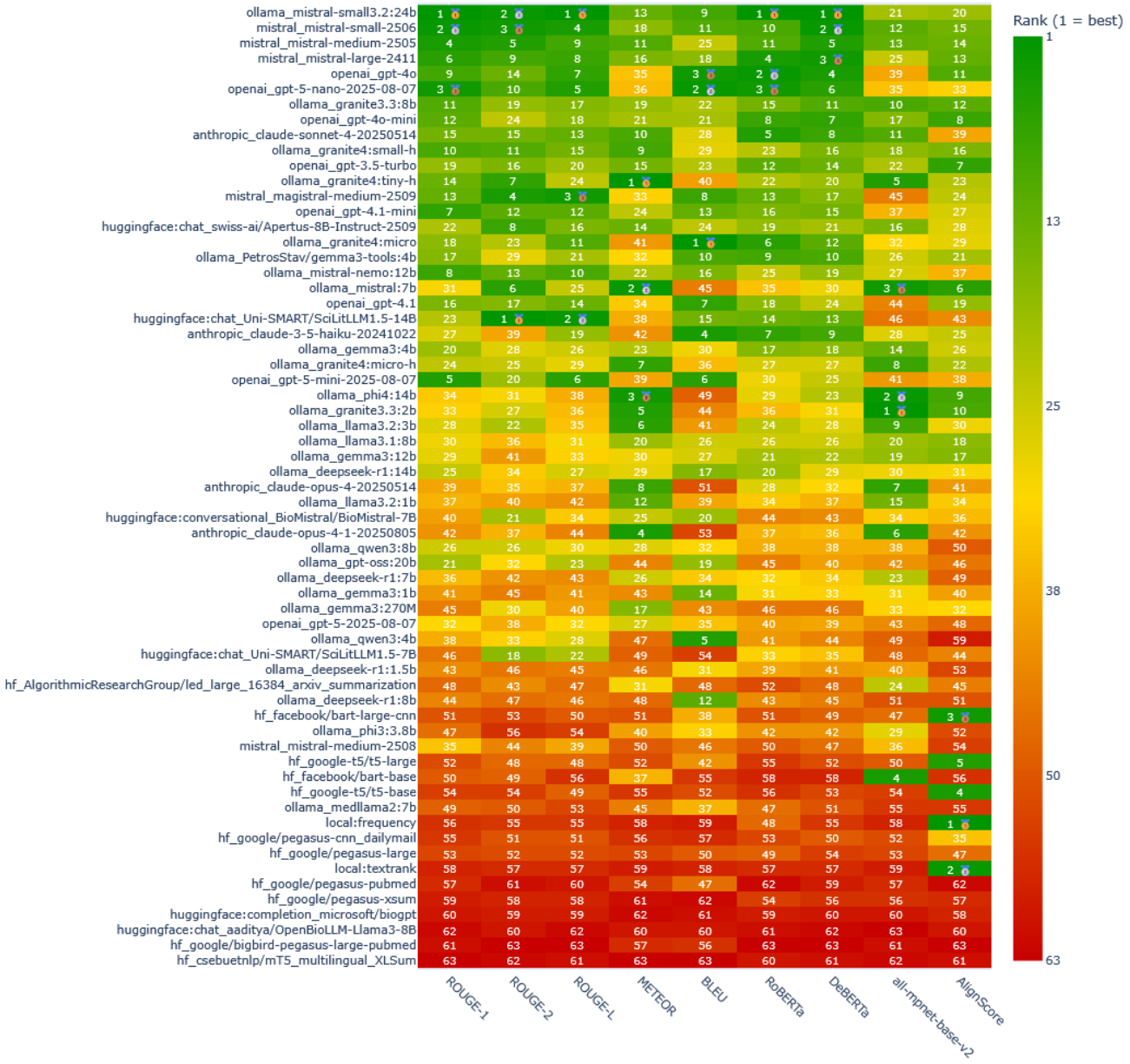


Figure 1. Model ranks across all surface-level and embedding-based metrics. Each row represents a model and each column a metric, with lower ranks indicating better performance. Models are ordered by their average rank across metrics.

3.2. Group Comparisons

Figure 2 summarizes the average performance of the five model categories based on the overall Metric Mean Score. General-purpose LLMs achieved the highest mean score (0.521), followed closely by reasoning-oriented LLMs (0.508). Encoder–decoder models and traditional extractive methods performed considerably lower, with mean scores of 0.445 and 0.451, respectively. The specialized models group showed the weakest overall performance (0.432).

Figure 3a compares the two largest and most competitive groups (general-purpose and reasoning-oriented LLMs) across multiple evaluation aspects. General-purpose models outperformed reasoning-oriented models in all measured categories, including surface-

level metrics, embedding-based metrics, execution time, compliance with word-length bounds, and overall Metric Mean Score. The largest advantage was observed in execution time, where general-purpose LLMs produced summaries more efficiently on average. Figure 3b provides a more detailed view of execution time differences between the two groups. The difference in quality metrics was smaller but consistent, with slightly higher scores across both surface-level and embedding-based metrics for general-purpose LLMs.

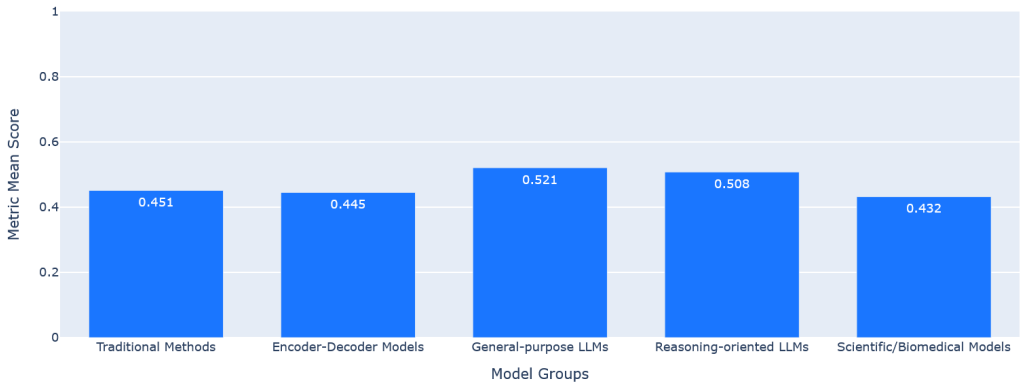
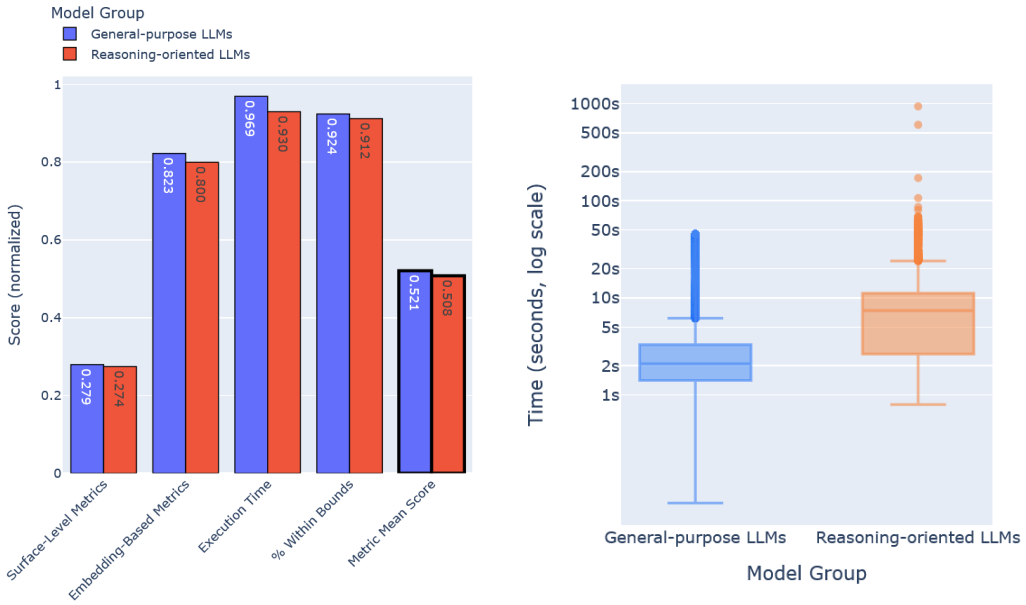


Figure 2. Average Metric Mean Score across the five model categories. General-purpose LLMs achieved the highest mean score, followed by reasoning-oriented LLMs. Traditional, encoder–decoder, and specialized models performed considerably lower.



(**a**) Comparison between general-purpose and reasoning-oriented LLMs across multiple evaluation aspects.

(**b**) Distribution of execution times for general-purpose and reasoning-oriented LLMs.

Figure 3. (**a**) Comparison between general-purpose and reasoning-oriented LLMs across multiple evaluation aspects. (**b**) Distribution of execution times for the same two groups. General-purpose models outperformed reasoning-oriented models across all measured categories and achieved lower, more stable runtimes.

3.3. Metric Correlations

To examine how the different evaluation metrics relate to each other, we computed pairwise Pearson correlation coefficients across all models (Figure 4). Each cell in the

matrix represents the correlation between two metrics based on their mean scores over all evaluated methods.

Strong positive correlations were observed among the surface-level metrics (ROUGE-1, ROUGE-2, ROUGE-L, METEOR, and BLEU). ROUGE variants were almost identical in their behavior ($\rho > 0.9$), while BLEU and METEOR showed slightly weaker but still substantial alignment with the ROUGE measures.

Most embedding-based metrics (RoBERTa, DeBERTa, and all-mpnet-base-v2) also showed very high internal consistency ($\rho > 0.8$), which reflects their shared focus on semantic similarity beyond surface-level overlap. When compared with the surface-level metrics, correlations were moderate to strong ($\rho \approx 0.7$–1.0), indicating that the two categories capture related but not identical dimensions of summary quality.

AlignScore correlated only moderately with the other metrics ($\rho \approx 0.4$–0.7), which can be attributed to its different point of reference, as it compares generated summaries directly with the source abstracts instead of the reference summaries used by the other metrics.

Overall, these relationships show that the various metrics are broadly consistent while still providing complementary perspectives. This supports the use of an aggregated "Metrics Mean Score" as a balanced indicator of overall summarization performance.

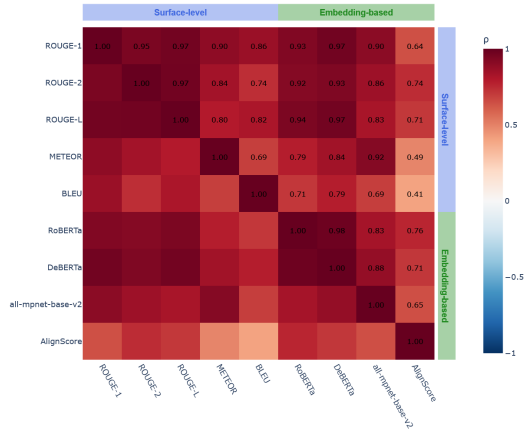


Figure 4. Correlation matrix of all evaluation metrics. Each cell represents the Pearson correlation coefficient (ρ) between two metrics based on their mean scores across models. Surface-level and most embedding-based metrics show strong internal consistency, while AlignScore exhibits lower correlations due to its distinct focus on factual consistency with the source abstracts.

3.4. Maybe include -> Performance by Metric Category

- show the performance when using only subsets of the metrics (only surface-level metrics vs only embedding-based metrics)

3.5. Maybe include -> Compliance with Summary Length

- not sure if this is worth its own subsection but it is interesting to see as it also gives a good feeling for how well a model follows the given instructions.

3.6. Maybe include -> Runtime Performance

- present the execution time across models, including distribution and outliers to give more context on execution time than just the average time for each model.

3.7. Maybe include -> Other things to possibly include

- maybe show a handpicked example of a good generated summary (good scores across all/most metrics, coming from a top-performing model) and a bad summary (bad scores across all/most metrics, coming from a low-performing model)

4. Discussion

5. Conclusion

6. Results

This section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation as well as the experimental conclusions that can be drawn.

6.1. Subsection

6.1.1. Subsubsection

Bulleted lists look like this:

- First bullet;
- Second bullet;
- Third bullet.

Numbered lists can be added as follows:

1. First item;
2. Second item;
3. Third item.

The text continues here.

6.2. Figures, Tables and Schemes

All figures and tables should be cited in the main text as Figure 5, Table 3, etc.

Figure 5. This is a figure. Schemes follow the same formatting.

Table 3. This is a table caption. Tables should be placed in the main text near to the first time they are cited.

Title 1	Title 2	Title 3
Entry 1	Data	Data
Entry 2	Data	Data [1]

[1] Tables may have a footer.

The text continues here (Figure 6 and Table 4).

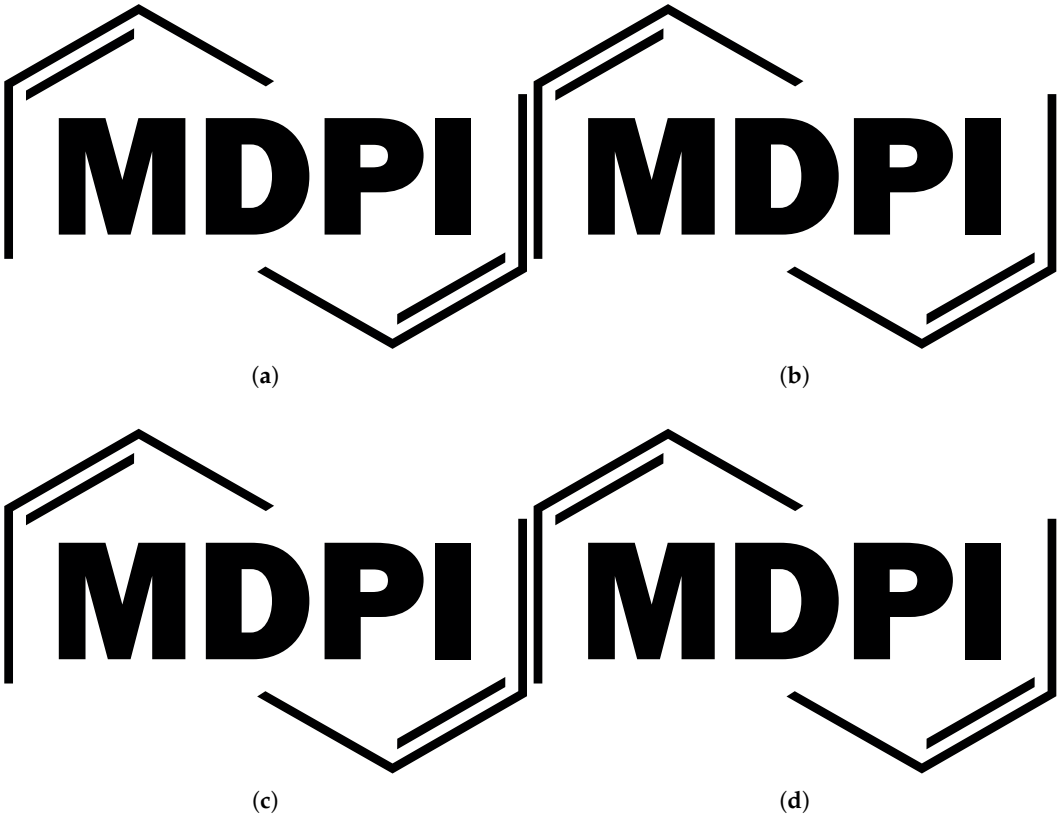


Figure 6. This is a wide figure. Schemes follow the same formatting. If there are multiple panels, they should be listed as: (**a**) Description of what is contained in the first panel. (**b**) Description of what is contained in the second panel. (**c**) Description of what is contained in the third panel. (**d**) Description of what is contained in the fourth panel. Figures should be placed in the main text near to the first time they are cited. A caption on a single line should be centered.

Table 4. This is a wide table.

Title 1	Title 2	Title 3	Title 4
Entry 1 *	Data Data Data	Data Data Data	Data Data Data
Entry 2	Data Data Data	Data Data Data	Data Data Data

* Tables may have a footer.

Text.

Text.

6.3. Formatting of Mathematical Components

This is the example 1 of equation:

$$a = 1, \tag{1}$$

the text following an equation need not be a new paragraph. Please punctuate equations as regular text.

This is the example 2 of equation:

$$a = b + c + d + e + f + g + h + i + j + k + l + m + n + o + p + q + r + s + t + u + v + w + x + y + z \quad (2)$$

Please punctuate equations as regular text. Theorem-type environments (including propositions, lemmas, corollaries etc.) can be formatted as follows:

Theorem 1. *Example text of a theorem.*

The text continues here. Proofs must be formatted as follows:

Proof of Theorem 1. Text of the proof. Note that the phrase "of Theorem 1" is optional if it is clear which theorem is being referred to. □

The text continues here.

7. Discussion

Authors should discuss the results and how they can be interpreted from the perspective of previous studies and of the working hypotheses. The findings and their implications should be discussed in the broadest context possible. Future research directions may also be highlighted.

8. Conclusions

This section is not mandatory, but can be added to the manuscript if the discussion is unusually long or complex.

9. Patents

This section is not mandatory, but may be added if there are patents resulting from the work reported in this manuscript.

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used "Conceptualization, X.X. and Y.Y.; methodology, X.X.; software, X.X.; validation, X.X., Y.Y. and Z.Z.; formal analysis, X.X.; investigation, X.X.; resources, X.X.; data curation, X.X.; writing—original draft preparation, X.X.; writing—review and editing, X.X.; visualization, X.X.; supervision, X.X.; project administration, X.X.; funding acquisition, Y.Y. All authors have read and agreed to the published version of the manuscript.", please turn to the CRediT taxonomy for the term explanation. Authorship must be limited to those who have contributed substantially to the work reported.

Funding: Please add: "This research received no external funding" or "This research was funded by NAME OF FUNDER grant number XXX." and and "The APC was funded by XXX". Check carefully that the details given are accurate and use the standard spelling of funding agency names at https://search.crossref.org/funding, any errors may affect your future funding.

Institutional Review Board Statement: In this section, you should add the Institutional Review Board Statement and approval number, if relevant to your study. You might choose to exclude this statement if the study did not require ethical approval. Please note that the Editorial Office might ask you for further information. Please add "The study was conducted in accordance with the Declaration of Helsinki, and approved by the Institutional Review Board (or Ethics Committee) of NAME OF INSTITUTE (protocol code XXX and date of approval)." for studies involving humans. OR "The animal study protocol was approved by the Institutional Review Board (or Ethics Committee) of NAME OF INSTITUTE (protocol code XXX and date of approval)." for studies involving animals. OR "Ethical review and approval were waived for this study due to REASON (please provide a detailed justification)." OR "Not applicable" for studies not involving humans or animals.

Informed Consent Statement: Any research article describing a study involving humans should contain this statement. Please add "Informed consent was obtained from all subjects involved in the study." OR "Patient consent was waived due to REASON (please provide a detailed justification)." OR "Not applicable" for studies not involving humans. You might also choose to exclude this statement if the study did not involve humans.

Written informed consent for publication must be obtained from participating patients who can be identified (including by the patients themselves). Please state "Written informed consent has been obtained from the patient(s) to publish this paper" if applicable.

Data Availability Statement: We encourage all authors of articles published in MDPI journals to share their research data. In this section, please provide details regarding where data supporting reported results can be found, including links to publicly archived datasets analyzed or generated during the study. Where no new data were created, or where data is unavailable due to privacy or ethical restrictions, a statement is still required. Suggested Data Availability Statements are available in section "MDPI Research Data Policies" at https://www.mdpi.com/ethics.

Acknowledgments: In this section you can acknowledge any support given which is not covered by the author contribution or funding sections. This may include administrative and technical support, or donations in kind (e.g., materials used for experiments). Where GenAI has been used for purposes such as generating text, data, or graphics, or for study design, data collection, analysis, or interpretation of data, please add "During the preparation of this manuscript/study, the author(s) used [tool name, version information] for the purposes of [description of use]. The authors have reviewed and edited the output and take full responsibility for the content of this publication."

Conflicts of Interest: Declare conflicts of interest or state "The authors declare no conflicts of interest." Authors must identify and declare any personal circumstances or interest that may be perceived as inappropriately influencing the representation or interpretation of reported research results. Any role of the funders in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; or in the decision to publish the results must be declared in this section. If there is no role, please state "The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results".

Abbreviations

The following abbreviations are used in this manuscript:

MDPI	Multidisciplinary Digital Publishing Institute
DOAJ	Directory of open access journals
TLA	Three letter acronym
LD	Linear dichroism

Appendix A

Appendix A.1

The appendix is an optional section that can contain details and data supplemental to the main text—for example, explanations of experimental details that would disrupt the flow of the main text but nonetheless remain crucial to understanding and reproducing the research shown; figures of replicates for experiments of which representative data are shown in the main text can be added here if brief, or as Supplementary Data. Mathematical proofs of results not central to the paper can be added as an appendix.

Table A1. This is a table caption.

Title 1	Title 2	Title 3
Entry 1	Data	Data
Entry 2	Data	Data

Appendix B

All appendix sections must be cited in the main text. In the appendices, Figures, Tables, etc. should be labeled, starting with "A"—e.g., Figure A1, Figure A2, etc.

References

1. Zhang, Y.; Jin, H.; Meng, D.; Wang, J.; Tan, J. A Comprehensive Survey on Process-Oriented Automatic Text Summarization with Exploration of LLM-Based Methods, 2025, [arXiv:cs.AI/2403.02901].
2. Zhang, H.; Yu, P.S.; Zhang, J. A Systematic Survey of Text Summarization: From Statistical Methods to Large Language Models, 2024, [arXiv:cs.CL/2406.11289].
3. Rohil, M.K.; Magotra, V. An exploratory study of automatic text summarization in biomedical and healthcare domain. *Healthcare Analytics* **2022**, *2*, 100058. https://doi.org/https://doi.org/10.1016/j.health.2022.100058.
4. Xie, Q.; Luo, Z.; Wang, B.; Ananiadou, S. A Survey for Biomedical Text Summarization: From Pre-trained to Large Language Models, 2023, [arXiv:cs.CL/2304.08763].
5. Luhn, H.P. The Automatic Creation of Literature Abstracts. *IBM J. Res. Dev.* **1958**, *2*, 159–165.
6. Edmundson, H.P. New Methods in Automatic Extracting. *J. ACM* **1969**, *16*, 264–285. https://doi.org/10.1145/321510.321519.
7. Robertson, S. Understanding Inverse Document Frequency: On Theoretical Arguments for IDF. *Journal of Documentation - J DOC* **2004**, *60*, 503–520. https://doi.org/10.1108/00220410410560582.
8. Reeve, L.H.; Han, H.; Nagori, S.V.; Yang, J.C.; Schwimmer, T.A.; Brooks, A.D. Concept frequency distribution in biomedical text summarization. In Proceedings of the Proceedings of the 15th ACM International Conference on Information and Knowledge Management, New York, NY, USA, 2006; CIKM '06, p. 604–611. https://doi.org/10.1145/1183614.1183701.
9. Mihalcea, R.; Tarau, P. TextRank: Bringing Order into Text. In Proceedings of the Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing; Lin, D.; Wu, D., Eds., Barcelona, Spain, 2004; pp. 404–411.
10. Shang, Y.; et al. Learning to rank-based gene summary extraction. *BMC Bioinformatics* **2014**, *15*, S10. https://doi.org/10.1186/1471-2105-15-S12-S10.
11. Erkan, G.; Radev, D.R. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research* **2004**, *22*, 457–479. https://doi.org/10.1613/jair.1523.
12. Afzal, M.; Alam, F.; Malik, K.M.; Malik, G.M. Clinical Context–Aware Biomedical Text Summarization Using Deep Neural Network: Model Development and Validation. *J Med Internet Res* **2020**, *22*, e19810. https://doi.org/10.2196/19810.
13. Almasoud, A.; Hassine, S.; Al-Wesabi, F.; Nour, M.; Hilal, A.; Al Duhayyim, M.; Hamza, A.; Motwakel, A. Automated Multi-Document Biomedical Text Summarization Using Deep Learning Model. *Computers, Materials & Continua* **2022**, *71*, 5800. https://doi.org/10.32604/cmc.2022.024556.
14. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need, 2023, [arXiv:cs.CL/1706.03762].
15. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2019, [arXiv:cs.CL/1810.04805].
16. Moradi, M.; Samwald, M. Clustering of Deep Contextualized Representations for Summarization of Biomedical Texts, 2019, [arXiv:cs.CL/1908.02286].
17. Liu, Y. Fine-tune BERT for Extractive Summarization, 2019, [arXiv:cs.CL/1903.10318].
18. Sefid, A.; Giles, C.L. SciBERTSUM: Extractive Summarization for Scientific Documents, 2022, [arXiv:cs.CL/2201.08495].
19. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. *CoRR* **2019**, *abs/1910.13461*, [1910.13461].
20. Yuan, H.; Yuan, Z.; Gan, R.; Zhang, J.; Xie, Y.; Yu, S. BioBART: Pretraining and Evaluation of A Biomedical Generative Language Model. In Proceedings of the Proceedings of the 21st Workshop on Biomedical Language Processing; Demner-Fushman, D.; Cohen, K.B.; Ananiadou, S.; Tsujii, J., Eds., Dublin, Ireland, 2022; pp. 97–109. https://doi.org/10.18653/v1/2022.bionlp-1.9.
21. Abinaya, S.; Vigil, M.; Keerthika, K.; Varshasri, R. Medical Text Summarization Using BART with LoRA-Based Parameter Efficient Fine Tuning **2024**.

22. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer, 2023, [arXiv:cs.LG/1910.10683].
23. Zhang, J.; Zhao, Y.; Saleh, M.; Liu, P.J. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization, 2020, [arXiv:cs.CL/1912.08777].
24. Zaheer, M.; Guruganesh, G.; Dubey, A.; Ainslie, J.; Alberti, C.; Ontanon, S.; Pham, P.; Ravula, A.; Wang, Q.; Yang, L.; et al. Big Bird: Transformers for Longer Sequences, 2021, [arXiv:cs.LG/2007.14062].
25. Beltagy, I.; Peters, M.E.; Cohan, A. Longformer: The Long-Document Transformer. *arXiv:2004.05150* **2020**.
26. Steblianko, O.; Shymkovych, V.; Kravets, P.; Novatskyi, A.; Shymkovych, L. Scientific article summarization model with unbounded input length. *Information, Computing and Intelligent systems* **2024**, pp. 150–158. https://doi.org/10.20535/2786-8729.5.2024.314724.
27. Plaat, A.; Wong, A.; Verberne, S.; Broekens, J.; van Stein, N.; Back, T. Multi-Step Reasoning with Large Language Models, a Survey, 2025, [arXiv:cs.AI/2407.11511].
28. Radford, A.; Narasimhan, K. Improving Language Understanding by Generative Pre-Training. 2018.
29. Turbitt, O.; Bevan, R.; Aboshokor, M. MDC at BioLaySumm Task 1: Evaluating GPT Models for Biomedical Lay Summarization. In Proceedings of the Proceedings of the 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks; Demner-fushman, D.; Ananiadou, S.; Cohen, K., Eds., Toronto, Canada, 2023; pp. 611–619. https://doi.org/10.18653/v1/2023.bionlp-1.65.
30. Bai, Y.; Kadavath, S.; Kundu, S.; Askell, A.; Kernion, J.; Jones, A.; Chen, A.; Goldie, A.; Mirhoseini, A.; McKinnon, C.; et al. Constitutional AI: Harmlessness from AI Feedback, 2022, [arXiv:cs.CL/2212.08073].
31. Team, G.; Kamath, A.; Ferret, J.; Pathak, S.; Vieillard, N.; Merhej, R.; Perrin, S.; Matejovicova, T.; Ramé, A.; Rivière, M.; et al. Gemma 3 Technical Report, 2025, [arXiv:cs.CL/2503.19786].
32. Li, S.; Huang, J.; Zhuang, J.; Shi, Y.; Cai, X.; Xu, M.; Wang, X.; Zhang, L.; Ke, G.; Cai, H. SciLitLLM: How to Adapt LLMs for Scientific Literature Understanding, 2025, [arXiv:cs.LG/2408.15545].
33. Wang, C.; Kantarcioglu, M. A Review of DeepSeek Models' Key Innovative Techniques, 2025, [arXiv:cs.LG/2503.11486].
34. Elsevier. Highlights, 2024. Accessed: 2025-08-07.
35. Cell Press. Final Submission: Other Components: Highlights, 2024. Accessed: 2025-08-07.
36. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* **2020**, *21*, 1–67.
37. Hasan, T.; Bhattacharjee, A.; Islam, M.S.; Mubasshir, K.; Li, Y.F.; Kang, Y.B.; Rahman, M.S.; Shahriyar, R. XL-Sum: Large-Scale Multilingual Abstractive Summarization for 44 Languages. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online, 2021; pp. 4693–4703.
38. Mishra, M.; Stallone, M.; Zhang, G.; Shen, Y.; Prasad, A.; Soria, A.M.; Merler, M.; Selvam, P.; Surendran, S.; Singh, S.; et al. Granite Code Models: A Family of Open Foundation Models for Code Intelligence, 2024, [arXiv:cs.AI/2405.04324].
39. Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. The Llama 3 Herd of Models, 2024, [arXiv:cs.AI/2407.21783].
40. Jiang, A.Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D.S.; de las Casas, D.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. Mistral 7B, 2023, [arXiv:cs.CL/2310.06825].
41. Abdin, M.; Aneja, J.; Awadalla, H.; Awadallah, A.; Awan, A.A.; Bach, N.; Bahree, A.; Bakhtiari, A.; Bao, J.; Behl, H.; et al. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone, 2024, [arXiv:cs.CL/2404.14219].
42. Abdin, M.; Aneja, J.; Behl, H.; Bubeck, S.; Eldan, R.; Gunasekar, S.; Harrison, M.; Hewett, R.J.; Javaheripi, M.; Kauffmann, P.; et al. Phi-4 Technical Report, 2024, [arXiv:cs.CL/2412.08905].
43. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners, 2020, [arXiv:cs.CL/2005.14165].
44. OpenAI.; Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; et al. GPT-4 Technical Report, 2024, [arXiv:cs.CL/2303.08774].
45. Anthropic. Claude - Models overview, 2025. Accessed: 2025-09-24.
46. Hernández-Cano, A.; Hägele, A.; Huang, A.H.; Romanou, A.; Solergibert, A.J.; Pasztor, B.; Messmer, B.; Garbaya, D.; Ďurech, E.F.; Hakimi, I.; et al. Apertus: Democratizing Open and Compliant LLMs for Global Language Environments. https://arxiv.org/abs/2509.14233, 2025.
47. DeepSeek-AI.; Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; et al. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning, 2025, [arXiv:cs.CL/2501.12948].
48. Team, Q. Qwen3 Technical Report, 2025, [arXiv:cs.CL/2505.09388].
49. OpenAI. GPT-OSS: Open Source GPT Models, 2025. Accessed: 2025-09-23.
50. OpenAI. GPT-5 Models, 2025. Accessed: 2025-09-24.

51. Mistral-AI.; :.; Rastogi, A.; Jiang, A.Q.; Lo, A.; Berrada, G.; Lample, G.; Rute, J.; Barmentlo, J.; Yadav, K.; et al. Magistral, 2025, [arXiv:cs.CL/2506.10910].
52. Luo, R.; Sun, L.; Xia, Y.; Qin, T.; Zhang, S.; Poon, H.; Liu, T.Y. BioGPT: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics* **2022**, *23*, [https://academic.oup.com/bib/article-pdf/23/6/bbac409/47144271/bbac409.pdf]. bbac409, https://doi.org/10.1093/bib/bbac409.
53. Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. Llama 2: Open Foundation and Fine-Tuned Chat Models, 2023, [arXiv:cs.CL/2307.09288].
54. Ankit Pal, M.S. OpenBioLLMs: Advancing Open-Source Large Language Models for Healthcare and Life Sciences. https://huggingface.co/aaditya/OpenBioLLM-Llama3-70B, 2024.
55. Labrak, Y.; Bazoge, A.; Morin, E.; Gourraud, P.A.; Rouvier, M.; Dufour, R. BioMistral: A Collection of Open-Source Pretrained Large Language Models for Medical Domains, 2024, [arXiv:cs.CL/2402.10373].
56. Li, S.; Huang, J.; Zhuang, J.; Shi, Y.; Cai, X.; Xu, M.; Wang, X.; Zhang, L.; Ke, G.; Cai, H. SciLitLLM: How to Adapt LLMs for Scientific Literature Understanding, 2024, [arXiv:cs.LG/2408.15545].
57. Lin, C.Y. ROUGE: A Package for Automatic Evaluation of Summaries. In Proceedings of the Text Summarization Branches Out, Barcelona, Spain, 2004; pp. 74–81.
58. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. Bleu: a Method for Automatic Evaluation of Machine Translation. In Proceedings of the Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics; Isabelle, P.; Charniak, E.; Lin, D., Eds., Philadelphia, Pennsylvania, USA, 2002; pp. 311–318. https://doi.org/10.3115/1073083.1073135.
59. Banerjee, S.; Lavie, A. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In Proceedings of the Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization; Goldstein, J.; Lavie, A.; Lin, C.Y.; Voss, C., Eds., Ann Arbor, Michigan, 2005; pp. 65–72.
60. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach, 2019, [arXiv:cs.CL/1907.11692].
61. He, P.; Liu, X.; Gao, J.; Chen, W. DeBERTa: Decoding-enhanced BERT with Disentangled Attention, 2021, [arXiv:cs.CL/2006.03654].
62. Song, K.; Tan, X.; Qin, T.; Lu, J.; Liu, T.Y. MPNet: Masked and Permuted Pre-training for Language Understanding, 2020, [arXiv:cs.CL/2004.09297].
63. Zha, Y.; Yang, Y.; Li, R.; Hu, Z. AlignScore: Evaluating Factual Consistency with a Unified Alignment Function, 2023, [arXiv:cs.CL/2305.16739].
64. McKinney, W. Data Structures for Statistical Computing in Python. In Proceedings of the Proceedings of the 9th Python in Science Conference; van der Walt, S.; Millman, J., Eds., 2010, pp. 51 – 56.
65. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.
66. Hagberg, A.; Swart, P.; S Chult, D. Exploring network structure, dynamics, and function using NetworkX. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.
67. Brin, S.; Page, L. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems* **1998**, *30*, 107–117. Proceedings of the Seventh International World Wide Web Conference, https://doi.org/https://doi.org/10.1016/S0169-7552(98)00110-X.
68. Bird, S.; Klein, E.; Loper, E. *Natural language processing with Python: analyzing text with the natural language toolkit*; " O'Reilly Media, Inc.", 2009.
69. Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K.Q.; Artzi, Y. BERTScore: Evaluating Text Generation with BERT, 2020, [arXiv:cs.CL/1904.09675].
70. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, 2019, [arXiv:cs.CL/1908.10084].
71. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. Transformers: State-of-the-Art Natural Language Processing. In Proceedings of the Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Online, 2020; pp. 38–45.